UNIVERSITY OF GUJRAT

UOG

A WORLD CLASS UNIVERSITY

Assignment NO: 02

Information Technology (Section B)

Session 2022-2026

Data Structures and Algorithms

Course Code: IT-121

Topic:

- **Program**

Submitted by:

Afshan Kanwal

22011556-093

Submitted to:

Mr. Ariz Mehmood

Date:

15th November, 2023

Complete this program:

- **Search**
- **update at any n point**
- **Insert at any n position**
- **Delete from beginning**
- **Delete from end**
- **Delete from any n position**
- **Search and update any point**

Program:

```
#include<iostream>
using namespace std;
class node {
public:
    int val;
    node* next;
    node(int data) {
        val = data;
        next = NULL;
    }
};
//insert in the start
node* inshead(node* &head, int val) {
    node* new_node = new node(val);
    new_node->next = head;
    head = new_node;
    return head;
}
```

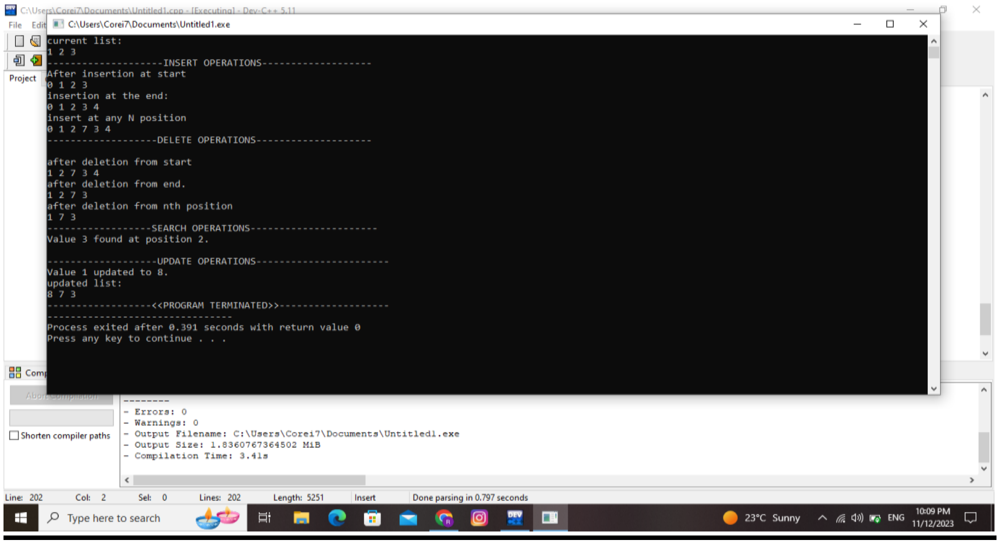
```
//insert in the end
node* insertAtEnd(node* &head, int val)
{
    node* new_node= new node(val);
    if(head==NULL){
    head=new_node;
}
node* temp=head;
while(temp->next!=NULL){
    temp=temp->next;
}
temp->next=new_node;
return head;
}
//insert at N position
node* insertatN(node* head, int val, int pos){
    if(pos==0){
        return inshead(head, val);
    }
    node* new_node=new node(val);
    node* temp=head;
    for(int i=0; i<pos-1 && temp!=NULL; ++i){
        temp=temp->next;
    }
    if(temp==NULL)
{
    cout<<"insertion at N is not possible "<<endl;
    delete new_node;
```

```
}
else{
    new_node->next= temp->next;
    temp->next= new_node;
}
return head;
}
//delete from start
node* del_start(node* &head){
    if(head==NULL)
{
    cout<<"deletion not possible "<<endl;
}
else {
    node* temp=head;
    head= temp->next;
    delete temp;
}
return head;
}
//delete from end
node* del_end(node* &head) {
    if (head == NULL) {
        cout << "Deletion not possible. " << endl;
    }
    else if (head->next == NULL) {
        delete head;
        head = NULL;
    }
    else {
        node* temp = head;
        while (temp->next->next != NULL) {
            temp = temp->next;
        }
        delete temp->next;
        temp->next = NULL;
    }
    return head;
}
//deletion from Nth position
node* del_atN(node* &head, int pos) {
    if (head == NULL) {
        cout << "Deletion not possible. List is empty." << endl;
    }
    else if (pos == 0) {
        return del_start(head);
    }
    else {
        node* temp = head;
        for (int i = 0; i < pos - 1 && temp != NULL; ++i) {
            temp = temp->next;
        }
        if (temp == NULL || temp->next == NULL) {
            cout << "Deletion at position " << pos << " is not possible." << endl;
        }
        else {
            node* toDelete = temp->next;
            temp->next = temp->next->next;
            delete toDelete;
        }
    }
    return head;
}
//search any value
node* searchValue(node* head, int val) {
    node* temp = head;
    int pos = 0;

    while (temp != NULL) {
        if (temp->val == val) {
            cout << "Value " << val << " found at position " << pos << "." << endl;
            return temp;
        }
        temp = temp->next;
        pos++;
    }

    cout << "Value " << val << " not found in the list." << endl;
    return NULL;
}
//to update values
void updateValue(node* head, int oldVal, int newVal) {
    node* temp = head;
    while (temp != NULL) {
        if (temp->val == oldVal) {
            temp->val = newVal;
            cout << "Value " << oldVal << " updated to " << newVal << "." << endl;
            return;
        }
        temp = temp->next;
    }

    cout << "Value " << oldVal << " not found in the list. Update failed." << endl;
}
//function t display elements
void displayList(node* head) {
    node* temp = head;
    while (temp != NULL) {
        cout << temp->val << " ";
        temp = temp->next;
    }
}
int main() {
    int val;
    node* existing = new node(1);
    existing->next = new node(2);
    existing->next->next = new node(3);
    cout<<"current list: "<<endl;
    displayList(existing);
    cout<<endl;
    cout<<"--------------------INSERT OPERATIONS-------------------"<<endl;
```

```
//inserting the value at the start
existing = inshead(existing, 0);
cout << "After insertion at start "<<endl;
displayList(existing);
existing= insertAtEnd(existing,4);
cout<<endl;
cout<<"insertion at the end: "<<endl;
displayList(existing);
//insertion at any N position
existing= insertatN(existing, 7,3);
cout<<endl;
cout<<"insert at any N position "<<endl;
displayList(existing);
cout<<endl;
cout<<"-------------------DELETE OPERATIONS--------------------"<<endl;
//deletion from start
existing= del_start(existing);
cout<<endl;
cout<<"after deletion from start "<<endl;
displayList(existing);
//deletion from end
existing= del_end(existing);
cout<<endl;
cout<<"after deletion from end. "<<endl;
displayList(existing);
//deletion from Nth position
existing= del_atN(existing, 1);
cout<<endl;
```

```
    cout<<"after deletion from nth position "<<endl;
    displayList(existing);
    cout<<endl;
    //search value
    cout<<"------------------SEARCH OPERATIONS---------------------"<<endl;
    node* result = searchValue(existing, 3);
    cout<<endl;
    //update value
    cout<<"------------------UPDATE OPERATIONS----------------------"<<endl;
    updateValue(existing, 1,8);
    cout<<"updated list: "<<endl;
    displayList(existing);
    cout<<endl;
    cout<<"------------------<<PROGRAM TERMINATED>>------------------";
    return 0;
}
```

Output:



C:\Users\Corei7\Documents\Untitled1.cpp - [Executing] - Dev-C++ 5.11
File
Edit
Project
C:\Users\Corei7\Documents\Untitled1.exe
Comp
Abort Compilation
□ Shorten compiler paths

```
current list:
1 2 3
--------------------INSERT OPERATIONS-------------------
After insertion at start
0 1 2 3
insertion at the end:
0 1 2 3 4
insert at any N position
0 1 2 7 3 4
-------------------DELETE OPERATIONS--------------------

after deletion from start
1 2 7 3 4
after deletion from end.
1 2 7 3
after deletion from nth position
1 7 3
------------------SEARCH OPERATIONS----------------------
Value 3 found at position 2.

------------------UPDATE OPERATIONS----------------------
Value 1 updated to 8.
updated list:
8 7 3
------------------<<PROGRAM TERMINATED>>------------------
--------------------------------
Process exited after 0.391 seconds with return value 0
Press any key to continue . . .
```

```
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Corei7\Documents\Untitled1.exe
- Output Size: 1.8360767364502 MiB
- Compilation Time: 3.41s
```